http://dx.doi.org/10.1109/IJCNN.2001.938835

Morrison, M.W., Dingle, A.A. and Attikiouzel, Y. (1996) A new approach to unsupervised Markov random field-based segmentation of Mr images. In: Fourth International Symposium on Signal Processing and Its Applications, ISSPA 96, 25 - 30 August, Gold Coast, Australia, pp. 2889-2892.

Algebraic Perceptron in Digital Channel Equalization

James P. Young, Thomas Hanselmann, Anthony Zaknich and Yianni Attikiouzel

Center for Intelligent Information Processing Systems (CIIPS)
Department of Electrical and Electronic Engineering
The University of Western Australia, Australia

Abstract

The paper investigates the application of the Algebraic Perceptron to solve the problem of channel equalization. The focus is on the particular case where the degree of intersymbol interference is severe. In recent years, some researchers have applied the Support Vector Machine for the same application and found valuable results. However, the Support Vector Machine requires solving a constrained optimization problem with quadratic programming, which is not a trivial task for large data sets. Like the Support Vector Machine, the Algebraic Perceptron also achieves linear separation in the high dimensional feature space, but with reduced calculation requirement. The tradeoff is that the separation surface is not a maximal margin one. In the simulation, it was found that for some channels the Algebraic Perceptron performed better than the Support Vector Machine. Further, given a more complete training set, the performance of the Algebraic Perceptron can match the performance of the Support Vector Machine.

1. Introduction

In channel equalization, conventional adaptive linear equalizers are fast, simple and effective solutions to linear channels with smooth spectral characteristics. If these conditions are removed, equalization will need to rely on non-linear equalizers. One of the prime requirements is that the equalizer must have low computation complexity. Many neural networks have been studied and have shown different strengths and weaknesses. The architectures for many of the non-linear equalizers grow to become unmanageably complex with increases in channel order and input dimensions. This is known as the *curse of dimensionality*. This problem is not apparent in the Support Vector Machine (SVM). In a recent paper, Sebald has published interesting results using the SVM to equalize nonlinear channels [1]. However, some problems still exist for the SVM to be truly practical for channel equalization. Finding a solution for the SVM requires a considerable amount of computation.

The Algebraic Perceptron (AP) is in principle very close to the SVM. The AP, as well, maps the training data for classification onto a high dimensional feature space to find an arbitrary separating hyperplane. However, unlike the SVM, the separation hyperplane is not one with maximal margin. This is the down side of the AP. The AP needs to be trained with a more complete data set to have equivalent performance as the SVM. On the other hand, the calculation intensity is in orders of magnitude lower than solving using the standard SVM.

Further, the AP has the ability to remove singularities in the training set. It was found in simulations that for solving certain common equalization problems where singularities and noises are severe, the AP has the advantage of speed and performance over the SVM.

2. Algebraic Perceptron

The implementation of AP involves two mathematical operations. The first is the nonlinear mapping of multi-dimensional input vectors to a high dimensional feature space. This is in accordance with Cover's theorem on the separability of patterns. Cover's theorem states that the multi-dimensional input space may be transformed into a new feature space where the patterns are linearly separable with high probability, provided the transformation is nonlinear and that the dimensionality of the feature space is high enough [2]. The transformation on to the feature space and operations in the feature space are only implied through the use of an inner product kernel. The inner product kernel must satisfy Mercer's theorem. Two of the most common ones are the polynomial kernel and the Radial-basis kernel. These are given in Table 1. These two types of kernel always satisfy Mercer's theorem [3]. This first operation is the similar to the SVM but with an additional procedure to lift the dimension and to normalize the length of the input vectors.

	Inner Product Kernel (K)
P-th order Polynomial	$(\mathbf{x}^T\mathbf{x}_i+1)^P$
Radial Basis Function	$\exp(-\frac{1}{2\sigma^2}\|\mathbf{x}-\mathbf{x}_i\|^2)$

Table 1: Inner Product Kernels which satisfy Mercer's Theorem

The second operation is to find a linear separation hyperplane in the feature space. This is done through the fundamental algebraic definition of inner products that when two vectors are orthogonal to each other, the inner product of the two vectors is zero.

2.1. Nonlinear Transformation to The Feature Space

The dimension of the input vector, $\mathbf{x} \in R^n$, is lifted to R^{n+1}. This is achieved by adding a constant value, λ, to each input vector. The length of each vector is then normalized to 1. This can be seen as mapping $\mathbf{x}$ to the unit-sphere $S\, R^{n+1} \subset R^{n+1}$. The lifted vectors on the unit-sphere are taken to the feature space, V, through the mapping $\phi : S \to V$.

The dimensionality of the feature space is purposely chosen to be large to satisfy Cover's theorem. The calculation complexity in the feature space is therefore implied. However, the potential calculation complexity in such high dimension can be avoided by the use of inner product kernel and the clever formulation of the problem.

The inner product kernel, K, translates two vectors in the lower dimension space, E, into inner products in V-space.

$$k(\mathbf{x},\mathbf{y}) = <\phi(\mathbf{x}),\phi(\mathbf{y})>_V = (<\mathbf{x},\mathbf{y}>_E)^p \qquad (1)$$

Where $\phi(.)$ is the transformation of a vector onto the feature space. However, $\phi(.)$ is never calculated directly.

2.2 Separation in Feature Space

For separable classes, when vectors on the unit sphere are multiplied by the respective desired output, $d \in \{\pm 1\}$, vectors of the two different classes are mapped on to the same hemisphere. Let us define $\mathbf{y} = \phi(\mathbf{x})d(\mathbf{x})$.

Vector $\mathbf{z}$ is a normal vector to the separating hyperplane. A violating vector, say $\mathbf{y}_i$, is a vector whose angle $\mathbf{z}_j$ has an angle greater than $\pi/2$, or the equivalently an inner product $<\mathbf{z}_j, \mathbf{y}_i> <0$.

The position of $\mathbf{z}$ is evaluated iteratively by the AP algorithm, starting from an arbitrary position. When separation has been achieved at iteration j, the separation plane of which $\mathbf{z}_j$ is normal to, defines the boundary of the hemisphere.

The AP algorithm is formulated as follows [4]:

1. Iteration j = 0, select an arbitrary initial orthogonal vector $\mathbf{z}_j$.
2. While $\exists\, \iota$ such that $<\mathbf{z}_j, \mathbf{y}_\iota> < 0$, do
3. $\mathbf{y}_i = \arg\{\min_{y_\iota} \{\mathbf{z}_j, \mathbf{y}_\iota\}\}$
4. $\mathbf{z}_{j+1} = \mathbf{z}_j - 2 <\mathbf{z}_j, \mathbf{y}_i> \mathbf{y}_i$, j=j+1.

Where

$<\mathbf{z}_j, \mathbf{y}_i> \mathbf{y}_i$ is the projection of vector $\mathbf{z}_j$ onto vector $\mathbf{y}_i$.

The above steps can be expressed recursively as follows:

$$\mathbf{z}_j = \mathbf{z}_0 - 2\sum_{i=1}^{j} <\mathbf{z}_{i-1},\mathbf{y}_i>_v \mathbf{y}_i \qquad (2)$$

To avoid evaluating vectors in the feature space, the algorithm can be restated in terms of inner products via kernel functions.

$$\begin{aligned} <z_j,\phi(x_l)>_V &= <z_0,\phi(x_l)>_V - 2\sum_{i=1}^{j} <z_{i-1},y_i>_V <y_i,\phi(x_l)>_V \\ &= k(z_0,x_l) - 2\sum_{i=1}^{j} <z_{i-1},x_{I(i)}>_V d^2(x_{I(i)})k(x_{I(i)},x_l) \\ &= k(z_0,x_l) - 2\sum_{i=1}^{j} <z_{i-1},x_{I(i)}>_V k(x_{I(i)},x_l) \qquad (3) \end{aligned}$$

The final $\mathbf{z}_j$ is expressed as a linear combination of the most violating vectors gathered at all iterations.

2.3 Classification

During classification, input vectors are first translated to the unit-sphere in the feature space as described in the procedures above. The decision criterion is the sign of the inner product between the test vector and $\mathbf{z}_j$ in the feature space. Positive signs indicate that the test vectors are labeled +1, and the inverse for the other classification.

$$d(\mathbf{x}) = sign <\phi(\mathbf{x}),\mathbf{z}_j>_V \qquad (4)$$

2.4 Violation Limit

The frequency of violations for each violating vector is tracked. Let us define a parameter called the violation limit, VL. When the frequency of violations for a vector is above this limit, the vector is expelled from the training set and the network is retrained. This vector can be regarded as a problematic one causing the algorithm not to converge. This violation limit caters for the non-separable classes. With this violation limit, the algorithm is guaranteed to converge to a solution. This is a desirable function if a solution is required where the dimensionality of the feature space is constrained. This is further discussed in section 3.1.

2.5 Training With Large Data Sets

Training large data set requires significantly more memory spaces or more kernel evaluations. To effectively utilize the memory space, the training data may be split into smaller chunks. Training of each chunk can be done in sequence.

When a separation is achieved in the initial chunk, the initial vector, $\mathbf{z}_0$, and the set of violation vectors are added to the next chunk of training sets. Training can begin again with the same initial boundary vector, $\mathbf{z}_0$. Such a training method achieves similar results while being very efficient.

3. Algebraic Perceptron Equalizer

Consider a sequence $\{\mathbf{S_n}\}$ is being transmitted through a dispersive channel $\mathbf{h}$ with transfer function

$$H(z) = \sum_{i=0}^{n_h} h_i z^{-i} \qquad (5)$$

The transmitted symbol sequence $\{\mathbf{S_n}\}$ is assumed to be an equiprobable and independent binary sequence taking values from $\{\pm 1\}$. The output is $\hat{\mathbf{y}} = \mathbf{S_n} * \mathbf{h}$. The output of the channel is further corrupted by zero mean Gaussian noise $\mathbf{v}$. The input sequence at the receiver end is $\mathbf{y} = \hat{\mathbf{y}} + \mathbf{v}$. The last m channel outputs are fed into an equalizer to recover the original transmitted symbols, or a delayed and/or phase shifted version of it.

Consider a channel with a transfer function

$$H(z) = 0.3482 + 0.8704z^{-1} + 0.3482z^{-2} \qquad (6)$$

Taking an equalizer input dimension of two, the possible combinations of input vectors are listed in Table 2. Figure 1 shows a sample of input vectors corrupted by noise at 20dB. With the delay of 1, the crosses and the circles represent input vectors belonging to the two classes respectively.

The equalization task can now be treated as classification problem, mapping $x \in R^n$ to $y \in R$. Given the training data pairs and the desired value being the delayed input to the channel.

	s(t)	s(t-1)	s(t-2)	s(t-3)	y(t)	y(t-1)
1	1	1	1	1	1.5668	1.5668
2	1	1	1	-1	1.5668	0.8704
3	1	1	-1	1	0.8704	-0.174
4	1	1	-1	-1	0.8704	-0.8704
5	1	-1	1	1	-0.174	0.8704
6	1	-1	1	-1	-0.174	0.174
7	1	-1	-1	1	-0.8704	-0.8704
8	1	-1	-1	-1	-0.8704	-1.5668
9	-1	1	1	1	0.8704	1.5668
10	-1	1	1	-1	0.8704	0.8704
11	-1	1	-1	1	0.174	-0.174
12	-1	1	-1	-1	0.174	-0.8704
13	-1	-1	1	1	-0.8704	0.8704
14	-1	-1	1	-1	-0.8704	0.174
15	-1	-1	-1	1	-1.5668	-0.8704
16	-1	-1	-1	-1	-1.5668	-1.5668

Table 2:Possible Combinations of Input Vectors to the Equalizer

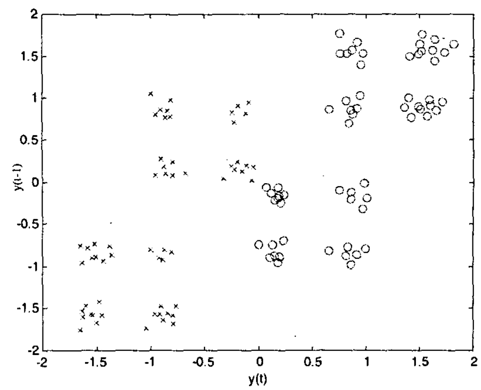

Figure 1: Plot of the Input Vectors to the Equalizer Corrupted by Noise

3.1. Singularities

For certain channels with severe ISI, the formulation described above for acquisition of training pairs may map the same input to different outputs. More information therefore is required for a definite separation. With the addition of noise, the equalizer is required to separate two classes of scattered input vectors around the same point. In this situation, where a definite separation cannot be achieved, it may be better in terms of overall performance to identify the singularities and remove them. For it is meaningless trying to generalize the effect of noise. By restricting the dimensionality of the feature space, inseparable points can be removed.

4. Simulation Results

The performances of the equalizers were tested on two channels using Monte Carlo simulation. The first channel has transfer function

$$H(z) = 0.227 + 0.460z^{-1} + 0.688z^{-2} + 0.460z^{-3} + 0.227z^{-4} \qquad (7)$$

There were severe non-separable points in the training set. The amplitude spectrum for this channel shows the presence of deep spectral null. Such channels are sometimes encountered in radio transmission [5]. The results of the AP were compared with SVM^{light}, RBFN and a conventional adaptive linear equalizer. The dimension of the input vectors for AP, SVM^{light} and RBFN were taken as 5.

The RBFN in this case required quite a large number of centers. The complexity was too large for practical purposes. Both training and evaluation of data required a considerable amount of time on a sequential machine. The performance of RBFN was simply used as a comparison.

The results of SVM were acquired using SVM^{light} [6]. This is a software package for solving the standard SVM especially crafted to handle large data sets. The parameter, C, which controls the tradeoff between complexity of the network and the number of non-separable points were adjusted experimentally to achieve the best result. In the simulation, the value of C was 500 for training data set of 1000. Convergence was a problem when the value of C was too small. The polynomial kernel was used for both the SVM and the AP. The power of the polynomial, p, was set to 10. The violation limit for the AP was set to 20. The result is shown in figure 2. In this case, the AP was the best performer.

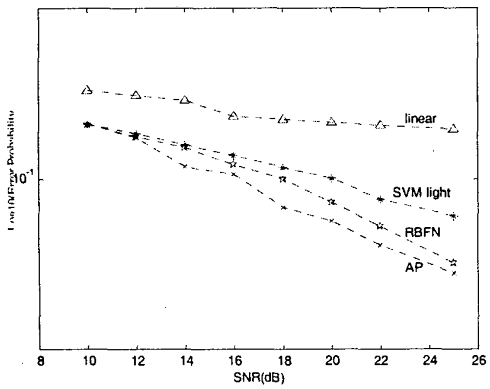

Figure 2: Performance of Equalizers for Severe ISI

For channels with no singularities, the AP suffered from not being a maximal margin classifier. When trained with small data sets, the performance of the AP fluctuated between different training data sets. However with larger training sets, the performance of the AP was close to what could be achieved with the SVM. The transfer function for the channel tested is given in equation (6).

The Wiener filter was used as a benchmark. The Wiener filter performance is the best result that a linear filter can achieve. The results of the SVM were acquired after training only with 1000 data set. The AP was trained with 10,000 data points and achieved a similar performance. The polynomial kernel was used for both the SVM and the AP with p=3.

5. Conclusion

The AP is a fast and efficient method for finding a linear separating plane in the feature space. The calculation is in orders of magnitude lower than solving the standard SVM. For channels with deep spectral null, the AP achieved better results. The explanation lies in the impossibility to definitely separate points, the separating plane achieved by a maximal margin classifier did not offer a better generalization. For less severe channels, the AP required a more complete training data to achieve the equivalent performance of the SVM.

Both the AP and the SVM were shown to be able to equalize high order channels, having as well high dimensional input vectors. This is the advantage over other nonlinear equalizers.

The AP removes high frequent violating points. These points indicate non-separable, or close to non-separable points, which form the singularities in the equalization problem. Therefore, by removing the most frequent violating points, the data set is smoothed and avoids modeling the noise, which results in improved generalization. In other signal processing applications, the AP may be used to preprocess given data.

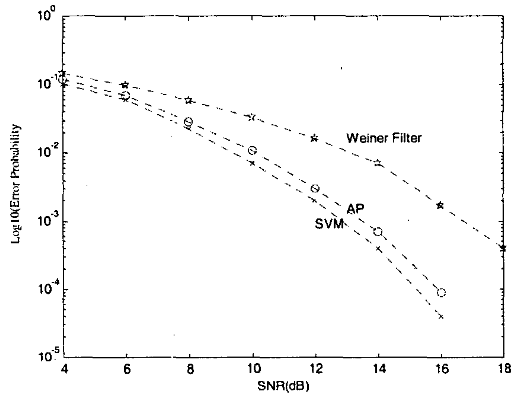

Figure 3: The Performance of Equalizers for mild ISI. AP was trained with a large data set

Acknowledgement

The authors wish to thank Dr. L.Noakes for the discussion on the mathematical aspects of Algebraic Perceptron. Thanks to Thorsten Joachims for publishing the code of SVM^{light}.

6. References

[1] D.J.Sebald, "Support Vector Machine Techniques for Nonlinear Equalization", *IEEE Trans. On Signal Processing*, Vol.48, No.11, Nov. 2000.

[2] T.M.Cover,"Geometrical and Statistical Properties of Systems and Linear Inequalities with Aplication in Pattern Recognition", *IEEE Trans. On Electronic Computers*, Vol.EC-14, pp326-334, 1965.

[3] S.Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd Ed, Prentice-Hall, 1999

[4] T.Hanselmann and L.Noakes, "Optimizing an Algebraic Perceptron Solution", *IJCNN*, 2001.

[5] J.G.Proakis, *Digital Communications*, 3rd Ed., McGraw-Hill, 1995.

[6] T.Joachims, software package SVM^{light}, version 3.5,2000. {http://ais.gmd.de/~thorsten/svm_light}